

ADVANCED PROJECTS

Build a website navigation toolbar

A navigation toolbar should be clear and simple. Jon Thompson explains how to make one using Cascading Style Sheets

Cascading Style Sheets (CSS) are a very useful tool for web developers. We'll show you how to use them to generate a navigation bar with drop-down menus to go on your website, but CSS can be used for a lot more. This tutorial will demonstrate the basic principles behind CSS and how to use it.

Each CSS style is called a selector. All selectors go in the <HEAD> section of the webpage. The simplest selector type is the tag selector. This redefines the appearance of standard HTML tags. Use a redefined tag and your browser will display the style as defined in the stylesheet instead of using the browser's default setting.

Here's a full example that changes the tag so that rather than making any subsequent text appear in bold, it appears in a red, 14-point Arial font:

```
<HTML>
<HEAD>
  <style type="text/css">
    B {
      font-family: arial;
      font-size: 14px;
      color: red;
    }
  </style>
</HEAD>
<BODY>
  <B>This is a customized version of
  bold</B>
</BODY>
</HTML>
```

The stylesheet is defined in the <style> section. This is something your browser understands implicitly. If you type the example into a plain-text file and save it with

the extension .html, loading it into your browser should produce red text.

The format for a selector definition is always the name followed by the properties you want to set, each separated by semicolons and surrounded by curly brackets. Each property in turn has a name and a value, separated by a colon and terminated by a semicolon.

Instead of having to copy and paste changes to your styles into the head section of every page on your site, you can split the CSS out into its own file and make a reference to it in the head section of each page. This makes global changes very easy:

```
<link rel=stylesheet href="mystyles.css"
type="text/css">
```

Change mystyles.css to match the URL for your own stylesheet.

A TOUCH OF CLASS

The class selector is more flexible than the tag selector. This defines a style as before but, instead of applying a blanket redefinition to all instances of a tag, you can apply a class selector to individual instances of tags using the class= attribute. This means you can apply a style to some paragraphs but not to others.

The browser needs to know when you are using a tag selector (such as <p>, the standard HTML tag for a paragraph) and a class selector that you have decided to invent. You can create a style called newbold in the stylesheet by adding an initial period to the class selector name. Here's an example:

```
<HTML>
<HEAD>
```

INTRODUCTION

Professional websites usually have a nice-looking navigation toolbar to help visitors find their way around the site. It can be tricky to create a toolbar that looks good while remaining flexible. Some designers use images as buttons, but a text-based navigation is more accessible and easier to update. Sadly, plain text doesn't look exciting.

With Cascading Style Sheets (CSS), you can have the best of both worlds, and make a toolbar with great-looking text-based buttons.

Jon Thompson

IT Correspondent



contributors@computershopper.co.uk

Back on track Adding background graphics

Instead of using text-based menus with coloured backgrounds, you can easily specify background images instead. The basic syntax is as follows:

```
background-image: url('button-background.
gif');
```

This reads the file called button-background.gif from the current directory. The image is displayed from the top left corner and is cropped if it extends beyond the bottom or right of the area.

You can easily make a fetching gradient for a pull-down menu. Create a gradient in your favourite painting program. It should be a few pixels wide and as long as the depth of the menu behind which it appears. Don't worry if it's too long, as it will be cropped. Set it as the background image and repeat it horizontally, left to right, like this:

```
background-image: url(gradient-
background.gif);
```



▲ The finished toolbar: CSS makes it easy to construct functional navigation bars without complex scripting

```
background-repeat: repeat-x;
```

The repeat-x property ensures that the image is displayed repeatedly for the exact width of the menu. You can do the same vertically, repeating images from the top to the bottom of an area to create a gradient that fades horizontally.

In this case, create a gradient that is a few pixels high and that stretches the width of the menu. Use the repeat-y property instead of repeat-x.

```
<style type="text/css">
.newbold {
  font-family: arial;
  font-size: 14px;
  color: red
}
</style>
</HEAD>
<BODY>
  <b class="newbold">This is bold
  text using the newbold style</b>
  <br>
  <b>And this uses normal bold</b>
</BODY>
</HTML>
```

In this example, the style is called newbold. The initial period is in the definition to show the browser what type of selector it is. We've added a class= attribute to the first bold tag in the body of the page so it uses the newbold style. In the body we refer to newbold and not .newbold. The period exists only in the stylesheet, not in the HTML file.

What's exciting about CSS is that some of the individual selector properties can change the look of a page depending on events. So we can change the look of links depending on whether they've been visited, or even if the cursor is hovering over the link.

AT THE BAR

We can use what we've learned to build a functional navigation bar that you can add to each page of your website. It will have two top-level options, each of which has a drop-down menu containing three options. These will appear when you hover the mouse pointer over the associated top-level choice. As an extra touch, when you run the mouse pointer over each option and sub-option they will also change colour automatically.

Here's the finished implementation, which we'll explain shortly. We've added line numbers to the listing for ease of reference. If you want to copy this code into a file for your own use, ignore the line numbers or it won't work:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//
  DTD HTML 4.01//EN" "http://www.
  w3.org/TR/html4/strict.dtd">
2 <html>
3   <head>
4     <style type="text/css" >
5       .top-level-option {
6         font-family: arial;
7         text-align: center;
8         float: left;
9         width: 150px;
10        border: 1px solid;
11        margin: 1px;
12        background: #ffa912;
13        color: black;
14      }
15      .top-level-option: hover {
16        background: #99ff99;
17      }
18      .sub-menu-options {
19        display: none;
20        list-style: none;
21        border: 1px solid;
22        background: #99ffcc;
23        margin: 1px;
24        padding: 0px;
25      }
```

```
26      .top-level-option: hover ul {
27        display: block;
28      }
29      .sub-menu-options li: hover {
30        background: #f945a3;
31      }
32    </style>
33  </head>
34
35  <body>
36    <div class="top-level-option">
      Menu 1
37      <ul class="sub-menu-
        options">
38        <li><a href="#">Menu 1
          sub-option 1</a></li>
39        <li><a href="#">Menu 1
          sub-option 2</a></li>
40        <li><a href="#">Menu 1
          sub-option 3</a></li>
41      </ul>
42    </div>
43
44    <div class="top-level-
      option">Menu 2
45      <ul class="sub-menu-
        options">
46        <li><a href="#">Menu 2
          sub-option 1</a></li>
47        <li><a href="#">Menu 2
          sub-option 2</a></li>
48        <li><a href="#">Menu 2
          sub-option 3</a></li>
49      </ul>
50    </div>
51
52  </body>
53 </html>
```

Don't forget about line 1. If this isn't included in the file, Internet Explorer (IE) will ignore all our carefully crafted hover functionality.

Take a look at the body section of the listing, beginning at line 35. Each of the top-level options, which forms a line across the screen, sits within its own block. This block is defined by a pair of <div> and </div> tags, short for 'division'. There's one at line 36 and another at line 44. A div is simply a convenient way of grouping elements together. The two <div> blocks each contain a piece of text, Menu 1 and Menu 2. These are the names that will appear in the two top-level options.

Inside each <div> block, there's a further block, marked with a pair of tags, at lines 37 and 45. These are unordered lists, which contain the drop-down menus. Each menu contains a set of tags to define the options. There are three in each and they have names that reflect their position in the menu system, such as Menu 2 sub-option 3. Each list item has an associated hyperlink. To use these, you need to change the hash symbol to a valid URL on your site.

If you wish to add more top-level options to the screen, each with its own drop-down menu, copy and paste one of the <div> blocks and edit the lines.

STYLE CONSCIOUS

The top-level-option style begins at line 5. We apply this style to both top-level options by assigning its class to the <div> blocks that surround the options. This style first sets the font and the text alignment. The float property

ensures that the options display in a line from the left. If you want them to flow from the right, use float: right.

This style also sets the width, border, spacing and colour for each top-level option to which it is applied. In this case, each will be 150 pixels wide with a black border one pixel thick, with black text and an orange background.

When we move the mouse pointer over one of the top-level options, we want it to change its background colour temporarily to indicate that it's available to be selected. This is achieved with the top-level-option: hover style, beginning at line 15. This defines the action that the browser will take in response to a visitor hovering the mouse pointer over a top-level option. In this case, we want to make a temporary change to the background colour to provide the user with some visual feedback.

The :hover part of the selector is called a pseudo-class, and there's a small but useful set of these you can use to specify actions for the browser when different events occur. These include changing the look of an option, depending on whether its hyperlink has been visited or not. So you could, for instance, specify a text colour for links with the following tag selectors:

```
A:link {color: blue;}
A:visited {color: cyan;}
```

These definitions can instantly change any aspect of the presentation of the link, such as the background colour or font. See the 'Back on track' box opposite for more details, but bear in mind that not all the available pseudo-classes are implemented in IE yet.

At line 18, the sub-menu-option style begins. This is similar to the top-level-option style in that it changes the colour, border (and so on) of the unordered list that represents each drop-down menu. However, without the padding: 0px property at line 24, Firefox will add a chunk of space before each of the options in the drop-down menus, causing them to wrap. Setting this property makes sure that your menu bar looks identical in both IE and Firefox.

FEELING THE DROP

We need to keep the drop-down menus hidden from view until the mouse is over the associated top-level option. Hiding the menu is achieved with the display: none property at line 19. By default, HTML lists display bullet points, which we don't want, so line 20 gets rid of them.

How do we display the drop-down menus? This is the job of the top-level-option: hover ul style, which begins at line 26. The syntax for this selector looks complicated because it is defining an action for a specific condition. The selector is telling the browser: "Whenever the mouse hovers over a page element to which the top-level-option style applies, temporarily set the following properties for the unordered list it contains." The only property we set is to tell the browser to display the unordered list, thereby making the drop-down menu appear.

Finally, when the user runs the mouse pointer over the options on a drop-down menu, we want the background of the selected option to change. This is the job of the sub-menu-options li: hover selector, which adds an action for the browser to take when the mouse pointer hovers over an entry. 